Integrating Context Information into Enterprise Applications for the Mobile Workforce – A Case Study

A. Spriestersbach, SAP AG, H. Vogler, SAP Labs, F. Lehmann, T. Ziegert, TU-Dresden
{axel.spriestersbach, hartmut.vogler}@sap.com, {fl3, tz1}@inf.tu-dresden.de

ABSTRACT

The integration of context information (especially location information) into mobile applications and services is one of the most crucial requirements to achieve a broader usability and hence acceptance of these. So far location information is used for typical business-to-consumer applications such as mobile MapQuest or ATM-finder. The application of location awareness in typical enterprise or business applications, such as logistics or Customer Relationship Management (CRM), is currently addressed rather poor.

In this paper we discuss the enhancement of mobile enterprise applications by context information. Starting from a customer demand and for a mobile sales force scenario, our objective was the improvement of the usability of mobile enterprise applications by introducing context information to these.

Keywords

Context awareness, mobile computing, location awareness, enterprise applications

1. INTRODUCTION

Real-time access to enterprise data and applications for the mobile work force, such as sales representatives or service technicians, is a crucial factor for enterprises. As of today a lot of workflows especially for the employees on the road are still paper-based and therefore lack in automated processing and backend integration.

With new small, lightweight and inexpensive mobile devices as well as the increased coverage of wireless data networks mobile solutions have become applicable for a broad usage in the mobile work force. Beneath several technical problems such as the quality-limited connectivity, scare resources in terms of processing speed and memory capacity, ergonomics and usability are still the biggest barriers for a broad usage of mobile devices. The usability of small devices is worse compared to a well-equipped PC and depends on aspects like small display size, tiny and non-standard keyboards. New paradigms for data presentation and interacting with applications are therefore needed and leverage the usability on mobile devices.

In this paper we discuss the enhancement of mobile enterprise applications by context information. Starting from a customer demand and scenario for the mobile sales force, our objective was the improvement of the usability of mobile enterprise applications by introducing context information to these applications.

We are using a mySAP solution as backend enterprise software, thus no application logic has to be processed on the device. Additionally, we are using *SAP's Web Technology* [12], with special HTML or WML templates for mobile devices, as a middleware to connect the backend with the mobile devices. Since this technology is already a product used by several customers, we will not explain this approach in this paper in detail.

The second part of the paper, gives a brief overview of context information and related work. The third part describes the application scenario and the improvements by adding location awareness. The following section describes the realization of our approach. Finally we will conclude this paper and give a brief outlook to future work.

2. RELATED WORK AND DEFINITIONS

Context aware machines or applications are not a new concept, so various definitions for the term *context awareness* already exist. A definition used in [14] seems the most appropriate one. Therefore, the notion of context awareness for devices can be split up into three components: *activity*, *environment* and *self*. The *activity* component describes the task the user is working on and focuses therefore on the human using the device, and his or her habits. The *environment* describes the physical and social surroundings of the user. The current location, the activities in the environment and other extern properties like temperature or humidity belong to this. The *self* component contains the status of the device itself.

Location awareness as a subset of context awareness plays an important role, especially the derivation of implicit context information belonging to a particular location, plays an important role. In this paper context awareness should be used in terms of the location of the user and the device, respectively. Some applications already use location information. An example are

GPS (Global Positioning System, [4]) based navigation systems. These systems use physical location information.

Location awareness is determined to be part of the environment component of context awareness. Location awareness can be split up into two characteristics as defined in [10, 11].

Physical Location: Physical location specifies the location of an object based on a global coordinate system. An object location may be given by a set of coordinates such as a pair of latitude and longitude. This information can be provided with a varying degree of precision.

Semantic Location: A semantic location specifies the position of an object within a larger context. An example of such a context is a conference room, a shopping mall, a bus stop or the bus itself. Every context stores some additional information about a local environment and its resources.

There are several research efforts aiming to integrate context awareness in a web-based infrastructure. The joint objective of the following approaches is the augmentation of physical objects at physical places by additional, mostly web-hosted (context) information.

HPs Cooltown: The Cooltown approach [2, 3] is completely web-based. This means Cooltown uses URNs pointing to information located in the web. Therefore objects broadcast their identifier periodically. A mobile device uses so-called resolvers to get the respectively URL of the id. When the particular URL is found, the client is able to retrieve the information about the physical object.

Nexus: The Nexus Project [15] defines a Nexus world depicting the real world in a spatial model. The virtual world is populated for example with so called virtual information towers, which carry information about physical objects. This information may come from different sources (WWW, DBMS, etc.). So Nexus has a more general approach of augmenting a location in the real world with information stored in the virtual world.

Worldboard: Worldboard [13] uses a fairly abstract model for an augmented reality system and hence there are no dependencies to a specific technology, but therefore there is also no specific system design. The real world is divided into cubic meters or centimeters and users can attach pieces of information on every face of a cube.

The Cooltown approach is more specific and so the technical outcomes of this project (e.g. e-squirt) can be reused. Nexus tries to combine the general approach of Worldboard with the more technical approach of Cooltown. Therefore the system is more open but also more complex than Cooltown.

3. BUSINESS SCENARIO AND PROTOTYPE EVOLUTION

Based on the demands of a customer (a large European watch manufacturer) we improved and automated their current workflow for recording sales orders. Currently, the sales representatives of this company fill paper forms for each customer. These forms are sent to a secretarial service once a week, where the form data is manually entered and transferred to the backend system during a weekly batch job. Obviously the current process is inefficient, time consuming, expensive and error-prone.

To satisfy the request of the customer and to archive an evolution of their existing business process towards a context aware mobile business solution we proposed a browser-based solution that directly connects the SAP based backend system. Therefore we assume a network connection to the backend system (online scenario). Currently we realized the access to two backend applications - Sales Order Entry and Availability Check - via a mobile device.

As a first step of our evolutionary process for implementing a prototype, we implemented the scenario using WAP-phones. Since we target the European market we decided to use WAP phones due to the broad deployment of WAP phones and the area-wide GSM network coverage in Europe. Additionally, most sales representatives are already equipped with mobile phones, which minimizes the rollout costs.

Due to the restricted user interface the application uses a reduced set of parameters compared to the paper based approach. The missing parameters are replaced with default values.

Figure 1: Sales Order Entry on WAP phone

The user interface of the sales order application shown in Figure 1 consists of a 6 digit customer number, a 6 digit products number and the quantity that is usually one or two digits, so the average amount of digits is 13.5 digits. The average time for entering a number on a WAP phone is 1,48 seconds ([1]). The overall time for entering one order is about 21 seconds on a usual WAP phone. This takes too long and is too inconvenient for the salesman.

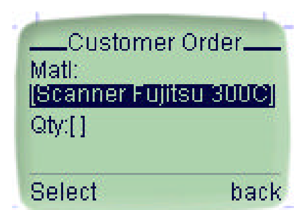

Figure 2: Context aware Sales Order Entry

The improved approach shown in Figure 2 uses contextual information to further reduce the manual input. It uses the current location of the sales representative in combination with customer addresses stored the backend to compute the customer number. The location information is derived from infrared transmitters placed at the customers side, but could also be derived from various other sources, e.g. GPS or MLS (see [8]).

Furthermore contextual information could be used to replace the product number input field by a selection list of appropriate products (see Figure 2). Product lists are usually too large for displaying them on a mobile phone. The contextual information allows the reduction of these lists based on context information and customer data stored in the backend, e.g. CRM system containing customers preferences, or supplied with the customers context, e.g. the customer demand for certain products.

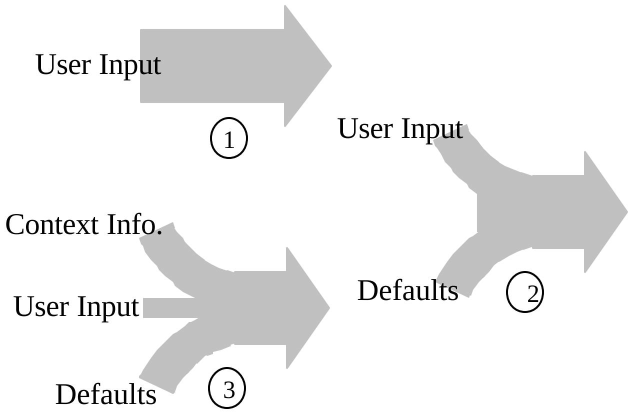

Figure 3: Prototype Evolution

Figure 3 summarizes the evolution from paper forms towards a context aware mobile solution. (1) illustrates the initial process that required the full amount of user interaction. The first mobile application illustrated in (2) decreases the necessary interaction by reducing the user interface. The deleted parts are replaced with default values. The utilization of contextual data to further reduce user interaction is illustrated in (3). User interaction is replaced with contextual or backend information. In addition default inputs of step (2) are optimized for the current context.

4. IMPLEMENTATION DETAILS

In our implementation we are using small infrared transmitters, which are installed at the customer's side, to retrieve location information. These transmitters called beacons broadcast a short message that contains an ID identifying the customer. A directory service, called *Resolver*, maps the ID to a URL that points to a XML document containing a corresponding virtual customer description. The URL is send to the *Sales Backend* application that generates an adapted web page (*Sales Application*) using the information stored in the XML document in combination with information from the backend. The application is displayed in the device browser to the sales representative.

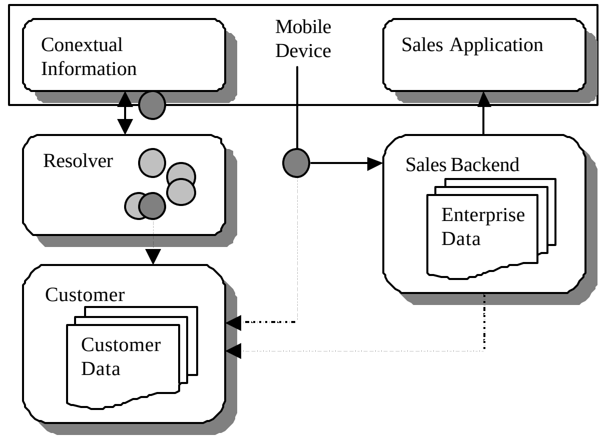

Figure 4 Architectural Overview

For receiving the beacons the approach relies on a small footprint installed on the device. The *Contextual Information Module* resolves all incoming IDs and transmits the resulting URLs to the backend. The usage of other network based localization information would lead to a zero footprint approach.

The indirect mapping of IDs and URLs, incorporating a directory service, enables changes regarding the deployed location technology. If GPS information should be used, another Resolver would be chosen to translate GPS coordinates into the URL of the place. The Resolver is also very valuable for the pre-processing and may direct to an appropriate XML document based on further contextual information. For instance there might be different roles the user holds when entering a shop. In our example of a shop, a customer entering the shop wants to buy a watch, thus she gets information about special offers etc. This approach also allows us to integrate different types of contextual information for other applications scenarios. Therefore the user has the ability to decide what will happen with the contextual data gathered by the device. This extends the usage of the Contextual Information Module by supporting different behavior depending on the selected role.

4.1 Technical Requirements and target platform

The implementation of our prototype implies the following technical requirements:

- The target device needs to be programmable to implement the sensing software.
- Wireless network connectivity, preferable GSM based.
- Directed sensing technology, e.g. infrared [5].
- HTML/WML browser on the mobile device.

Since ordinary cell phones are not programmable and so-called Smart phones, such as the Nokia Communicator 9210, currently are not widespread and available, these devices are not useful.

The *Palm Computing Platform* [10, 11] currently lacks in supporting wireless network connectivity outside the US and serial/IR connections in parallel are impossible.

Windows CE [7] as target platform meets all requirements. Thus we used a Compaq iPAQ H3600 in combination with a GSM-modem.

4.2 Software Architecture

The solution consists of the beacon, its counterpart the receiver and the built-in browser on the mobile device that displays the information provided by the *Receiver* module. A GSM module (see Figure 5) provides network connection.

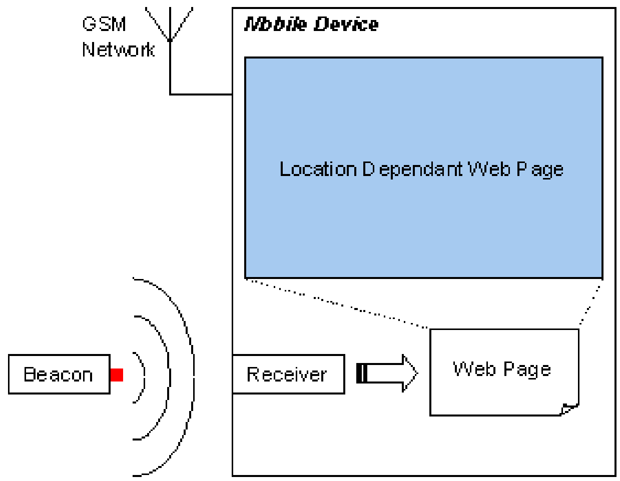

Figure 5: Beacon and Mobile Device with Receiver

4.2.1 Beacon Software

The beacon was implemented using a second WinCE device. The implementation is optimized for a small and lightweight footprint on the device.

The beacon searches for other IR devices within range and if it was successful, the beacon tries to transmit a short message. This message consists of a XML tag and the ID as its attribute (e.g. "`<beacon id="ID"/>`").

Any other IrDA compatible device capable of broadcasting small messages would be sufficient as beacon (e.g. [6]), although we used a powerful WinCE device.

4.2.2 Receiver – The Contextual Information Module

The *Receiver* is a service on the mobile device that receives incoming messages, resolves them to URLs and composes a local homepage. The homepage is displayed to the user upon request. The implementation of a Receiver is divided into three modules (shown in Figure 6) implemented as separated threads:

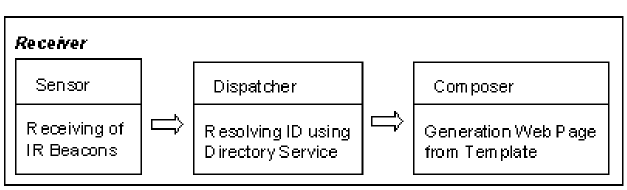

Figure 6: Internal Structure within Receiver Module

The *Sensor* listens for incoming beacons on the infrared port. After receiving a beacon, the URN is extracted and sent to the Dispatcher. To provide other contextual information other Sensor modules could be integrated. Depending on the type of contextual information, environmental sensors such as temperature sensors, light sensors or acceleration sensors can be integrated.

The *Dispatcher* queries a web based directory service (*Resolver*) for the URL corresponding to the received ID. We used HTTP GET requests and a servlet for resolving the ID. The URL points to an XML resource describing the shop or more general a place and is passed to the composer. Future versions the Dispatcher have to process the different types of contextual information before they are sent to the Resolver or bypassed to the Composer

The *Composer* creates the default homepage from local information and information found in the place's virtual XML description. This description consists of an URL to the place's default homepage, the type[1] of the entity described, resources and services available for that place and other useful information about the place. We expanded the description with customer numbers for suppliers.

Our approach differs here from the CoolTown concepts, which does not generate pages on the device. In Cooltown Beacons are sending URLs pointing to virtual representations that are displayed to the user. The shop's web server supplies the customization of these virtual descriptions. Therefore every user would need to provide sufficient credentials to get a customized version of this homepage.

In our scenario the virtual representation of the shop – the shops homepage – would have to include all sales order applications for different sales representatives that are visiting the shop. This is virtually impossible because the service functions the sales representatives are using are usually located in some backend system that might not be accessible from the outside. An alternative approach would be to add a link to the sales order application, but this could not be maintained since every sales representative would need an individual link and even minor changes to backend system's URL would result in updates to shop's web presentation. Additionally it is much more convenient for the sales representative to accesses his backend system always using the *same* link and different additional location dependent data instead of using various shop dependent links. With this approach we also achieve a more general solution for the originally defined problem, which opens a larger spectrum of applications beside sales orders. So in our approach the Receiver on the mobile device decides based on credentials on the current user role, which information to use from the place's XML description. For instance, during the week the sales representative would enter a

[1] Since most of this infrastructure is based on CoolTown, three types of entities are possible: people, things and places. Within this scenario at this point of time only places are needed and supported.

store to sell products, but he may also enter the store as a usual customer. The credentials or roles are bound to a set of templates that tell the composer how to create the default homepage using the location data.

The sales order application itself is rendered in the backend system. Therefore one template includes a HTTP request to the backend that also transmits a reference to the shop's virtual description. The reference is used to access the shops XML description and to extract data necessary to adapt the sales order application. For instance the customer number is used for identifying the customer and the preparation of a product list from a CRM system.

For other scenarios it could be necessary that the device itself needs some information about the place. To limit the data to be transferred to and from the mobile device, the composer does not request the whole XML description of the place, but makes an XQL query to the XML description. The information returned from the XQL query can be used to create the default homepage on the mobile device.

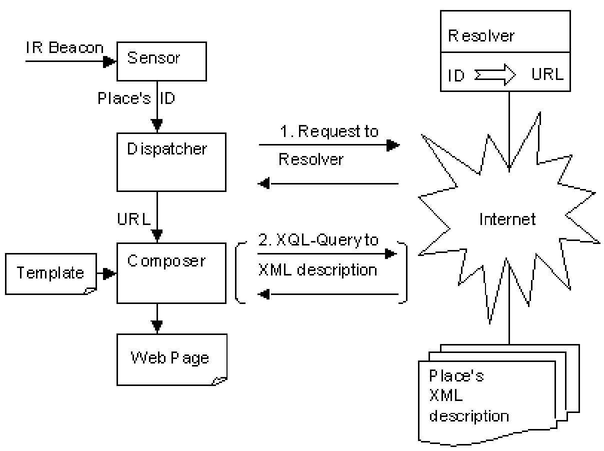

Figure 7: Receiver

Relying on a standard XML document to describe a place makes the approach very flexible. It is very easy to add more data to a place description.

5. CONCLUSION AND FUTURE WORK

In this paper we demonstrated the improvement of the usability for mobile enterprise application by adding location awareness. Since different sources of contextual information have to be combined (e.g. profile information of the backend and the portal as well as location information), the objective of a zero footprint on the device was not achieved (the prototype for the mobile device has a footprint of 22kByte). With location information provided by the wireless network we can achieve this goal.

Based on our prototype, we figured out, that adding context information to enterprise applications is not a generic approach. Different enterprise applications need different kinds of context information or at least a different interpretation. E.g. context information for a sales representative and context information for a customer has to be interpret in a different way.

In our future work we will investigate the usage of other wireless technologies such as Bluetooth [9] as well as other mobile devices such as cell phones for our application scenarios. Based on these experiences we will integrate location information into other typical mobile enterprise applications (e.g. service management). Additional usability tests will improve our current approach. Furthermore we will investigate the usage of other types of context information such as time or temperature.

6. REFERENCES

[1] O. Frick, A. Schmidt:WAP - Designing for Small User Interfaces, http://www.sapdesignguild.org/ index.asp?b=/innovation/artikel/wap-design1.html &t=/innovation/ innovation_topnav.html, 2000

[2] Hewlett-Packard Laboratories: Location Awareness in HP's CoolTown, http://www.cooltown.com/papers/ LocationAwareness.htm, 2000

[3] Hewlett-Packard Laboratories: Semantic Location, http://www.cooltown.com/papers/semantic/semantic.htm, 2000

[4] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins: Global Positioning System: Theory and Practice, Springer-Verlag, 1997

[5] Infrared Data Association: Minimal IrDA Protocol Implementation, http://www.irda.org/standards/pubs/ litever10.pdf, 1996

[6] Lesswire AG: LocalNavigator IrDA Beacon, http://www.lesswire.com/english/irda/content.htm, 2001

[7] Microsoft Corporation: Windows CE, http://www. microsoft.com/windows/embedded/ce/default.asp, 2001

[8] Mobile Location Services, MLS, http://www. gsmworld.com/technology/mls_presentations.html

[9] Official Bluetooth SIG Website, http://www.bluetooth.com, 2001

[10] Palm, Inc.: Palm OS, http://www.palmos.com/, 2001

[11] Palm, Inc.: Palm.com Products, http://www.palm.com/products/, 2001

[12] SAP AG: ITS and SAP@Web Studio Documentation, http://www.sapmarkets.com/its/itsdoc.htm, 2001

[13] J. C. Spohrer: Information in places, IBM Systems Journal, Vol 38, No. 4 - Pervasive Computing, 1999

[14] TEA. Technology for Enabling Awareness, http://tea.starlab.net/tea_vis.html, 2000

[15] S. Volz, D. Fritsch, D. Klinec, A. Leonhardi, J. Schützner: "NEXUS - Spatial Model Servers for Location-Aware Applications on the Basis of ArcView", Proceedings of the 14th ESRI European User Conference, Munich, 1999